

## WE CLAIM:

1. A method of decoding a packed representation of multiple parses comprising the steps of:

providing a packed representation including at least one  
5 edgenode, each edgenode including a substitution list;

creating a current forest object;

replicating the current forest object for each edgenode having a  
substitution list containing greater than one edgenode; and

traversing each edgenode of the packed representation.

2. The method of claim 1 further comprising the step of performing  
a scalar update of the current forest object for each edgenode having a  
substitution list containing exactly one edgenode.

3. The method of claim 1 further comprising the step of traversing  
each edgenode of the packed representation using a depth-first traversal.

4. The method of claim 1 wherein the current forest object is  
replicated by a number equal to a number of edgenodes in the substitution list  
20 of a current edgenode when the number of edgenodes in the substitution list  
is greater than one.

5. The method of claim 4 further comprising the step of performing  
a vector update of the current forest object with the substitution list of the  
25 current edgenode.

6. The method of claim 4 further comprising the step of updating each of the replicated forest objects with an element corresponding to a different edgenode in the substitution list.

5 7. The method of claim 1 further comprising the step of setting a current edgenode to a root edgenode of the packed representation.

8. The method of claim 7 further comprising the step of setting the current forest object to an empty forest object.

9. The method of claim 8 further comprising the step of setting a next edgenode of the packed representation to a leftmost child of the current edgenode.

10. A program for decoding a packed representation of parses stored on computer readable medium comprising the steps of:  
computer readable program code for creating a current forest object;  
computer readable program code for traversing each edgenode  
20 of the packed representation; and  
computer readable program code for replicating the forest object for each edgenode having a substitution list of elements greater than 1.

11. The program of claim 10 wherein the forest object is replicated a  
25 number of times equal to the number of elements in the substitution list.

12. The program of claim 11 further comprising computer readable code for performing a scalar update of the current forest object for each edgenode having a substitution list containing exactly one edgenode.

5 13. The program of claim 10 further comprising computer readable code for updating each of the replicated forests with a treenode corresponding to one of the elements in the substitution list.

10 14. The program of claim 10 further comprising computer readable code for setting a current edgenode to a root edgenode of the packed representation, and for setting the current forest object to an empty forest object.

15 15. The program of claim 14 further comprising:  
computer readable program for setting a next edgenode to a leftmost child of the current edgenode; and  
setting the current edgenode to the next edgenode.

20 16. The program of claim 15 further comprising  
computer readable program code for setting the next edgenode to a next one of the edgenodes in the substitution list when the current edgenode does not have at least one child.

25 17. The program of claim 16 further comprising  
computer readable program code for setting the current forest object to a forest object of the next one of the edgenodes in the substitution list.

18. The program of claim 17 further comprising  
computer readable program code for performing a node closure operation on  
a current forest object when the current edgenode in the last edgenode in the  
substitution list and when the current edgenode does not have a sibling to the  
right of the current edgenode.

19. The computer program of claim 18 further comprising  
computer readable program code for setting a next edgenode to null after the  
node closure operation, when the current edgenode does not have a parent.

20. The computer program of claim 18 further comprising  
computer readable program code for setting the current edgenode to parent of  
current edgenode, after the closure operation.

21. A system for decoding multiple parses comprising:  
a parser which receives output from a speech recognizer and  
creating parses stored in a packed representation, the packed representation  
including a plurality of edgenodes; each edgenode including a substitution list;  
an unpacking program stored on a computer readable medium  
including program code for creating an unpacked forest including the steps of  
creating a current forest object, traversing each edgenode of the packed  
representation using a depth-first traversal, replicating the current forest  
object a number of times equal to the number of edgenodes in the substitution  
list, and updating each copy of the current forest object with a treenode  
corresponding to one of the edgenodes of the substitution list.